(12) UK Patent Application (19) GB (11) 2 197 097 (13) A

(43) Application published 11 May 1988

(21) Application No 8626367

(22) Date of filing 4 Nov 1986

(71) Applicant
The University of Dundee

(Incorporated in United Kingdom)

Dundee DD1 4HN

(72) Inventors
Adrian Pickering
Andrew Swiffen

(74) Agent and/or Address for Service
Marks & Clerk,
57–60 Lincoln's Inn Fields, London WC2A 3LS

(51) INT CL⁴
G06F 7/00

(52) Domestic classification (Edition J):
G4A KS

(56) Documents cited
EP A2 0172357 US 4438505

(58) Field of search
G4A
Selected US specifications from IPC sub-class G06F

(54) Data entry using dynamic word predication algorithm

(57) A method of data input comprises establishing a database or vocabulary of words and entering a word prefix. The word prefix is used to select all words from the vocabulary commencing with that prefix to form a sub-set for subsequent processing. The subsequent processing involves ordering the members of the sub-set in accordance with their potential relevance and truncating the ordered sub-set so as to identify a predetermined number of the most potentially relevant words. The truncated ordered sub-set is subsequently presented to enable the intended input word to be selected, if present, from among the presented words.

The words are ordered in accordance with their recency of use, or their historical frequency of use, or preferably by a combination of the two.

The claims were filed later than the filing date within the period prescribed by Rule 25(1) of the Patents Rules 1982.
This print reflects (an) amendment(s) to the request for grant effected pursuant to Rule 35 of the Patents Rules 1982.

SPECIFICATION

Dynamic word prediction algorithm

This invention relates to an algorithm for dynamically developing a list of the most likely words that follow a given word prefix. The algorithm is adaptive in that it takes account of the past patterns of use of the words in developing its predictions. It has application in effort-efficient user interfaces in the field of Man-Machine interaction. In particular, the technique enables significant effort savings in computer interfaces for those whose physical abilities are impaired.

A 'word' is an ordered sequence of elements (normally characters) picked from an alphabet of elements bounded by word delimiters. A delimiter is normally an element from another set which is disjoint from the word alphabet set. A word prefix comprises the sub-sequences made up from zero or more of the first elements in a particular word. All the words together make up the vocabulary (set of all known words) from which higher order structures are built, such as sentences, which in turn make up a 'text'.

As words are chosen to make up sentences it is possible to record their usage statistics and make use of these to develop predictions of what is most likely to follow. The prediction subset can be refined every time further prefix information becomes available. They can be presented to the user so that a selection can be made from the subset in order to complete the word. To develop the entire subset of the vocabulary which applies at a particular instant is trivial. In practice, the cardinality of these subsets is so large that it is neither possible nor desirable to present the user with all their members. Thus it is necessary to order the set and truncate it so that the prediction list is assessable by the user and a member is selectable.

The algorithm presently to be described seeks to ensure that the the set ordering is optimal and only the most likely words are offered to the user. The member words that are not displayed remain accessible through use of the conventional input method, normally the QWERTY keyboard of a computer terminal. However, every time another element is added to the word prefix, the subset can be further refined. The selection device for choosing the predictions can be any of the wide variety available; examples being function keys on a conventional computer terminal, a touch-sensitive screen or the use of a 'mouse'.

A word w belongs to a vocabulary $V = \{w_1, w_2, \ldots, w_n\}$, containing n words. A text is an ordered sequence of t words in which words may be repeated e.g. $w_4\ w_1\ w_4\ w_{10} \ldots$ The text is fully defined by an index set $T = \{G_k | G_k \subset T\}$ where T is the set of all numbers from I. .t, k = I. .n and the disjoint sets G_k contain the word positions for w_k. The cardinality of G_k, $|G_k|$, corresponds to the number of times w_k was used. Typically the text demonstrates a Zipf distribution, expressed as:

$$\frac{|G_k|}{t} = \frac{\mu}{k}$$

where:

$$u = \frac{1}{\ln(n) + \gamma}$$

with y being Eulers' constant (~0.57721). Note that this is a very slow function of n when n >> 100. A rank ordering over V is denoted by k with w_1 being the highest frequency, lowest ranking word.

All members of V, w_k, have a corresponding average distance d_k between their use given by:

$$d_k = \frac{k}{\mu} \quad \text{words}$$

Thus, in order to capture all words that have a frequency rank position greater than some threshold value, k', text samples of $d_{k'}$ words must be examined. A special case of this is the average distance between the occurences of the rarest word w_{min} which must be the same as the text sample size. Thus a relationship exists between n and t:

$t = n[1n(n)+y]$

The Zipf distribution is an expression of the 'least cost' behaviour of humans when generating information carrying material when it is analysed a posteriori. It is essentially a static description and takes no account of the temporal behaviour of word selection; words fall into disuse and others increase in popularity as time proceeds and context changes.

The recency of a word can be expressed by its instantaneous word distance d'_k which is the span between the last position of w_k and the last word in the text (the current word). If the spans are measured in words, the values of d'_k are unique, lie in the range l. .t and are non-contiguous. The average of all the past d'_k spans will tend towards d_k.

Recency defines another ordering on V in terms of d'_k; the word w_x is said to be more recent than w_y if $d'_x < d'_y$. Studies have shown words in texts with Zipf-type distributions are more quickly found on average if they are examined in recency order. The 'move to front' strategy of self-organising lists and the cacheing principle are practical embodiments this principle used in other computing domains. In the context of prediction presentation, recency ordering provides a better basis on which to perform the set truncation.

The prediction subset V' is defined:

$$V' = \{w_k \mid w_k \subset V \text{ and all } w_k \text{ have a common prefix}\}$$

An ordering on V and thus on V' enables the first few members of V' to be chosen to be presented to the user. These form the set of predictions $P \subset V'$ such that $|P| \leqslant p$, with p being the maximum number of shown predictions.

Recency ordering does not perform so well when the d'_k are comparable with d_p since the members of P tend to be the same as the last few words in the text. This situation is encountered when trying to produce predictions with a zero initial prefix which yields a prediction subset of V. Here the historical ordering, based upon the frequency ordering of V, might provide a better basis for truncation. But, since the instantaneous distance d'_k is of greater importance for dynamic adaption, it may not be economic to maintain information relating to absolute word frequency as well. This frequency information might take the form of word usage counts or keeping a running average d_k. A discrete approximation sufficient for identifying the p most frequent words may be all that is necessary.

Each word in the vocabulary V has associated with it (a) recency information and, optionally, (b) frequency information. Both forms of information enable an ordering over V to be defined which may be used for developing the most suitable predictions P. Both need keeping up to date as the text being created is developed. Frequency information is relatively simple to update since only when an word is used is its frequency affected. However, the recency information is changing each time the text expands. If the number of words in the vocabulary, n, is large then updating the recency information can be impractical and the following methods may be employed to reduce the processing effort.

(1) The distance unit can be increased. In order that the d'_k are still unique the unit can be increased to that of the most frequent word. Assuming a Zipf distribution, the formula given above will give a minimum d_k of about 10 words.

Thus the recency information can be kept in units of 10 words and thus will only need updating after every tenth word. A further advantage of using a greater span unit is economy in the storage of the recency number.

(2) If a non-unique recency value is acceptable, then the span unit can be increased much further. Indeed the span can be such as to ensure that the p most regularly used words will always retain the highest recency. However, a problem with non-unique recency values is that, failing any further information, the truncation of V' to P can be somewhat arbitrary. However, if frequency information is available this can be used in conjunction with the recency information to resolve ties and produce a better P.

(3) The current value of the text size, t (in span units), can be written into the recency information field when the word is used. This can be done at the same time as the frequency information is updated. To derive d'_k this value is subtracted from t. Thus to derive the recency ordering for V' the d'_k are derived by subtracting the recency field from t at least each time t changes. This avoids having to explicitly run through the entire vocabulary after each span is complete updating the recency value.

The algorithm will now be presented by reference to an actual implementation embodied in a small, portable typing aid for those with motion disabilities. The microcomputer-based aid has provision for a vocabulary of up to 1000 words and the ability to display and use up to 5 predictions. The vocabulary is 'complete' as new words are captured and added when they occur for the first time. With $n = 1000$, $\mu = 1/7.485 = 0.1336$ and $t = 7485$ words by the above formulae. For efficiency reasons explicit updating of the recency values is performed. The average distance between the pth rank word, $d_p = 37.42$ words and thus the span between updating should well exceed this. A convenient span is the 150-200 words of a typical paragraph which provides a natural break in the text when updating can be performed. The frequency information provides the additional information for making the exact trucation position in V'. The coordinality of the recency information should allow the current 'working' vocabulary to be regarded as being recent. If 25% of the vocabulary is to be so treated then d_k for the 250th word is 1871 words or about 8-12 spans.

The frequency information is simply a word incidence count. Its size should be estimated by the formulae and allow for use well past the expected text size t. The highest expected count for the 1000 word vocabulary is 1000.

The ordering is performed by simple numeric comparison on the concatenation of recency and frequency treating the whole as an unsigned, binary coded number. Recency is represented using 1's complement so that the ordering is compatible with that of the word count, which is represented conventionally. In the specific example cited 15 bits are used to generate the ordering, identified b_0 to b_{14} with b_0 with lowest and b_{14} with highest binary weighting. The highest weighted 3 bits, b_{12} to b_{14}, are used for the recency representation and the remaining 12 bits, b_0 to b_{11}, are used to accommodate the count. The exact split between the available number of bits determines when the transition between recency and word count ordering occurs; minimum recency is represented by all zeros thus leaving the word count with its true weighting.

Other systems seeking to employ the ordering algorithm herein described should use the formulae and guidelines to produce an implementation to suit the specific circumstances.

CLAIMS

1. A method of data input comprising the steps of establishing a database or vocabulary of words (as hereinbefore defined);

entering a prefix of a word to be input;

establishing from the vocabulary a sub-set of words each of which includes the entered prefix, arranging the words of the sub-set in order of potential relevance;

truncating the ordered sub-set so as to limit the sub-set to a predetermined number of the most potentially relevant words; and

presenting the truncated ordered sub-set so that the word to be input may be selected if present in the truncated ordered sub-set.

2. A method of data input as claimed in Claim 1, wherein ordering of words of the sub-set includes determining the recency of use of each word of the sub-set.

3. A method of data input as claimed in Claim 1 or 2, wherein ordering of the sub-set includes determining the historical frequency of use of each of the words of the sub-set for a predetermined period of data input.

4. A method of data input as claimed in Claim 3, comprising the step of storing the absolute word frequency count for each word in the vocabulary so as to define said historical frequency of use.

5. A method of data input as claimed in Claim 3, comprising the step of storing a cumulative average for the frequency of use of each word of the vocabulary in order to determine said historical frequency of use.

6. A method of data input as claimed in Claim 2, comprising the step of storing an absolute recency value for each word of the vocabulary in order to define recency of use of each word.

7. A method of data input as claimed in Claim 2, comprising the steP of storing for each word of the vocabulary a recency value which is up-dated only after input of a plurality of words.

8. A method of data input as claimed in Claim 7, comprising the step of selecting said plurality to be equal to the number of words between subsequent occurrences of the most frequently used word for a predetermined period of data input.

9. A method of data input as claimed in Claim 2, comprising the step of storing for each word of the vocabulary the value of the total number of words input at the time the word was most recently used and subsequently subtracting said value from the current total number of words input in order to define the recency of use of the word for ordering of the sub-set.

10. A method of data input as claimed in any preceding Claim wherein the words are stored in binary digital form.

11. A method of data input as claimed in Claim 10 and claim 2 wherein the recency information for each word is stored in I's complement form.

12. A method of data input substantially as hereinbefore described.

13. Apparatus when adapted and arranged to carry out the method of any preceding claim.

Published 1988 at The Patent Office, State House, 66/71 High Holborn, London WC1R 4TP. Further copies may be obtained from The Patent Office, Sales Branch, St Mary Cray, Orpington, Kent BR5 3RD. Printed by Burgess & Son (Abingdon) Ltd. Con. 1/87.